



IN THE CLAIMS

1 1. [Cancelled]

1 2. [previously amended] A method according to claim 8, further comprising:

2 - when processing at least one captured data packet, determining

3 a modification command affecting said at least one captured data

4 packet, and

5 - maintaining a list of modification commands, said list enabling

6 modification of captured data packets.

1 3. [previously amended] A method according to claim 2, further comprising:

2 - modifying captured data packets based on said list of

3 modification commands, and

4 - releasing modified captured data packets.

1 4. [previously amended] A method according to claim 8, further comprising the

2 step of discarding captured data packets that are declined from processing.

1 5. [previously amended] A method according to claim 8, wherein captured data

2 packets that are declined from processing are delayed .

1 6. [amended herein] A method for handling data packets, said data packets
2 belonging to a set of data packets, said method comprising:
3 - capturing data packets,
4 - accepting a captured data packet for processing based on said
5 captured data packet and data packets captured prior to said captured
6 data packet,
7 - when processing at least one captured data packet accepted for
8 processing, determining a modification command affecting at least said
9 at least one captured data packet,
10 - maintaining a list of modification commands, said list enabling
11 modification of captured data packets,
12 - declining a captured data packet from processing, if said
13 captured data packet is already processed and modification commands
14 induced by said captured data packet are already determined and
15 maintained in said list of modification commands,
16 - modifying said captured data packet based on said list of
17 modification commands, and
18 - releasing the modified captured data packet.

1 7. [previously amended] A method according to claim 8, further comprising:
2 - declining a captured data packet from processing, if said

3 captured data packet is already processed, and

4 - releasing the captured data packet.

1 8. [previously amended] A method for handling data packets, said data
2 packets belonging to a set of data packets having at least partly hierarchical
3 structure, said method comprising:

4 - capturing data packets,

5 - accepting a captured data packet for processing or declining a
6 captured data packet from processing based on said captured data
7 packet and data packets captured prior to said captured data packet,

8 - said accepting comprising accepting data packets for
9 processing in the order specified by said at least partly hierarchical
10 structure

11 - declining a captured data packet from processing if a data
12 packet immediately preceding said captured data packet in said at least
13 partly hierarchical structure is not yet captured,

14 - delaying data packets declined from processing,

15 - accepting a delayed data packet for processing and processing
16 said delayed data packet, if data packet(s) immediately preceding said
17 delayed data packet in said at least partly hierarchical structure is(are)
18 already processed, and

19 - releasing the delayed data packet.

1 9. [previously amended] A method according to claim 8, comprising:

2 - accepting a captured data packet for processing, if data
3 packet(s) immediately preceding said captured data packet in said at
4 least partly hierarchical structure is(are) already processed.

1 10. [Cancelled]

1 11. [Cancelled]

1 12. [previously amended] A method according to claim 8, comprising

2 - delaying data packets declined from processing,
3 - accepting a captured data packet for processing,
4 - accepting delayed data packet(s) for processing, if data
5 packet(s) immediately preceding said delayed data packet(s) in said at
6 least partly hierarchical structure is(are) accepted for processing,
7 - processing said delayed data packet(s) together with said
8 captured data packet, and
9 - releasing the delayed and the captured data packets.

1 13. [previously amended] A method according to claim 8, wherein said at least
2 partly hierarchical structure is a sequence of data packets.

1 14. [previously amended] A method according to claim 8, wherein in that said
2 at least partly hierarchical structure is a hierarchically structured tree.

1 15. [Cancelled]

1 16. [previously amended] A method for handling data packets, said data
2 packets belonging to a set of data packets and forming a plurality of groups of
3 data packets, said method comprising:

4 - capturing data packets,

5 - declining a captured data packet belonging to a first group of
6 data packets from processing, if all other data packets belonging to said
7 first group of data packets are not yet captured, and delaying said
8 captured data packet, and

9 - processing a captured data packet belonging to a first group of
10 data packets together with delayed data packets belonging to said first
11 group of data packets if said captured data packet belonging to said first
12 group and said delayed data packets belonging to said first group form
13 a full first group of data packets.

1 17. [preveiously amended] A method according to claim 16, wherein said
2 plurality of groups of data packets further belong to a set of groups having at
3 least partly hierarchical structure, and further comprising
4 - processing said groups of captured data packets in the order
5 specified by said at least partly hierarchical structure.

1 18. [previously amended] A method according to claim 6, wherein said data
2 packets belonging to a set of data packets are first handled in a first node of a
3 cluster of network elements and said list of modification commands is
4 maintained in said first node, and in that said method further comprises the
5 step of:
6 - transmitting said list of modification commands from said first
7 node to a second node of said cluster of network elements.

1 19. [previously amended] A method according to claim 18, further
2 comprising:
3 - after said transmission of said list, handling said set of data
4 packets in said second node.

1 20. [previously amended] A method according to claim 19, further

2 comprising:

- 3 - when beginning to handle said data packets, storing in said first
4 node in a connection data structure an entry representing said set of
5 data packets, and
- 6 - before handling said set of data packets in said second node,
7 transmitting said entry from said first node to said second node.

1 21. [previously amended] A method according to claim 6 further comprising:

- 2 - defining a plurality of first pieces of information which are to be
3 replaced in the captured data packets with a plurality of corresponding
4 second pieces of information,

5 and wherein, in the processing of captured data packets,

- 6 - said first pieces of information are searched for, and
- 7 - if a first piece of information is found, at least one modification
8 command specifying at least the replacement of said first piece of
9 information with a corresponding second piece of information is
10 determined.

1 22. [previously amended] A method according to claim 21, wherein, if the
2 length of said first piece of information is different from the length of said
3 corresponding second piece of information and if said first piece of information

4 is found in payload of data packet(s), said modification command comprises
5 instructions for changing value of at least one header field in a data packet.

1 23. [previously amended] A method according to claim 6, wherein a
2 modification command comprises

3 - a first identifier indicating the beginning of a first piece of
4 information in the original captured data packets, the first piece of
5 information being subject to be replaced by a second piece of
6 information,

7 - the length of the first piece of information, and

8 - the second piece of information.

1 24. [previously amended] A method for handling data packets, said data
2 packets belonging to a set of data packets, said method comprising:

3 - capturing data packets,

4 - accepting a captured data packet for processing based on said
5 captured data packet and data packets captured prior to said captured
6 data packet,

7 - when processing at least one captured data packet accepted for
8 processing, determining a modification command affecting at least said
9 at least one captured data packet,

10 - maintaining a list of modification commands, said list enabling
11 modification of captured data packets,
12 wherein a modification command comprises
13 - a first identifier indicating the beginning of a first piece of
14 information in the original captured data packets, the first piece of
15 information being subject to be replaced by a second piece of
16 information,
17 - the length of the first piece of information,
18 - the second piece of information,
19 - a second identifier indicating the beginning of the second piece
20 of information in the modified captured data packets,
21 - an offset between a third identifier indicating the end of the first
22 piece of information in the original captured data packets and a fourth
23 identifier indicating the end of the second piece of information in the
24 modified captured data packets, and
25 - the length of the second piece of information.

1 25. [previously amended] A method according to claim 6, wherein said data
2 packets contain information fragments belonging to a sequence of information
3 fragments, said method further comprising the steps of:
4 - processing the information fragments of the captured data

5 packets in the order specified by said sequence.

1 26. [previously amended] A method according to claim 25, wherein each
2 information fragment of said sequence is processed only once.

1 27. [previously amended] A method for handling data packets, said data
2 packets belonging to a set of data packets, said method comprising:

3 - capturing data packets, said data packets containing information
4 fragments belonging to a sequence of information fragments,

5 - declining a captured data packet from processing if a data
6 packet containing the information fragment immediately preceding the
7 information fragments of said captured data packet in said sequence is
8 not yet captured, and

9 - accepting a captured data packet for processing, if a data
10 packet containing the information fragment immediately preceding the
11 information fragments of said captured data packet in said sequence is
12 already processed, whereby each information fragment of said
13 sequence is processed only once.

1 28. [previously amended] A method according to claim 25, wherein said
2 sequence of information fragments is a sequence of octets of data according

3 to the Transfer Control Protocol.

1 29. [previously amended] A software entity for handling data packets, said
2 data packets belonging to a set of data packets, said software entity
3 comprising

4 - program code means for capturing data packets,

5 - program code means for accepting a captured data packet for
6 processing or declining a captured data packet from processing based
7 on said captured data packet and data packets captured prior to said
8 captured data packet.

9 - program code means for handling said data packets belonging
10 to a set of data packets first in a first node of a cluster of network
11 elements and for maintaining said list of modification commands in
12 said first node, said list enabling modification of captured data packets,

13 - program code means for transmitting said list of modification
14 commands from said first node to a second node of a cluster of network
15 elements,

16 - program code means for declining a captured data packet from
17 processing if said captured data packet is already processed and
18 modification commands induced by said captured data packet are
19 already determined and maintained in said list of modification

20 commands,
21 - program code means for modifying said captured data packet
22 based on said list of modification commands, and
23 - program code means for releasing the modified captured data
24 packet.

1 30. [cancelled]

1 31. [previously amended] A software entity according to claim 29, further
2 comprising:

3 - program code means for modifying captured data packets
4 based on said list of modification commands, and
5 - program code means for releasing modified captured data
6 packets.

1 32. [amended herein] A software entity for processing data packets, said
2 data packets belonging to a set of data packets, said software entity being
3 adapted to receive data packets, and said software entity comprising:
4 - program code means for capturing data packets,
5 - program code means for accepting a captured data packet for
6 processing or declining a captured data packet from processing based

7 on said captured data packet and data packets captured prior to said
8 captured data packet

9 - program code means for processing received data packets first
10 in a first node of a cluster of network nodes, if said data packets
11 belonging belong to said a set of data packets ~~first in a first node of a~~
12 ~~cluster of network elements,~~

13 - program code means for determining a modification command
14 affecting at least received data, as a response to processing said data,
15 and

16 - said software entity being adapted to output said modification
17 command,

18 - program code means for maintaining said list of modification
19 commands in said first node, said list enabling modification of captured
20 data packets,

21 - program code means for transmitting said list of modification
22 commands from said first node to a second node of said cluster of
23 network elements,

24 - program code means for processing said set of data packets in
25 said second node after said transmission of said list of modification
26 commands from said first node to said second node,

27 - program code means for declining a captured data packet from

28 processing if said captured data packet is already processed and
29 modification commands induced by said captured data packet are
30 already determined and maintained in said list of modification
31 commands,
32 - program code means for modifying said captured data packet
33 based on said list of modification commands, and
34 - program code means for releasing the modified captured data
35 packet.

1 33. [cancelled]

1 34. [cancelled]

1 35. [previously amended] A network element for handling data packets in a
2 cluster of network elements, said data packets belonging to a set of data
3 packets, said network element comprising
4 - means for capturing data packets,
5 - means for accepting a captured data packet for processing or
6 declining a captured data packet from processing based on said
7 captured data packet and data packets captured prior to said captured
8 data packet,

9 - means for maintaining a list of modification commands, said list
10 enabling modification of captured data packets, and
11 - means for transmitting said list of modification commands to a
12 second network element of said cluster of network elements,
13 - means for declining a captured data packet from processing if
14 said captured data packet is already processed and modification
15 commands induced by said captured data packet are already
16 determined and maintained in said list of modification commands,
17 - means for modifying said captured data packet based on said
18 list of modification commands, and
19 - means for releasing the modified captured data packet.

1 36. [previously amended] A network element according to claim 35, further
2 comprising

3 - means for processing a captured data packet, and
4 - means for determining a modification command affecting at
5 least one captured data packet as a response to processing said at
6 least one captured data packet.

1 37. [previously amended] A network element cluster for handling data
2 packets, said data packets belonging to a set of data packets, at least one

3 node of said network element cluster comprising:

4 - means for capturing data packets,

5 - means for accepting a captured data packet for processing or

6 declining a captured data packet from processing based on said

7 captured data packet and data packets captured prior to said captured

8 data packet,

9 - means for maintaining a list of modification commands, said list

10 enabling modification of captured data packets,

11 - means for transmitting said list of modification commands from

12 said node to another node of said cluster of network elements,

13 - means for declining a captured data packet from processing, if

14 said captured data packet is already processed and modification

15 commands induced by said captured data packet are already

16 determined and maintained in said list of modification commands,

17 - means for modifying said captured data packet based on said

18 list of modification commands, and

19 - means for releasing the modified captured data packet.

1 38. [cancelled]

1 39. [previously amended] A storage medium carrying a computer-

executable software entity for handling data packets, said data packets
belonging to a set of data packets, said software entity comprising

- program code configured to capture data packets,
- program code configured to accept a captured data packet for
processing based on said captured data packet and data packets
captured prior to said captured data packet,
- program code configured to, when processing at least one
captured data packet accepted for processing, determine a modification
command affecting at least said at least one captured data packet,
- program code configured to maintain a list of modification
commands, said list enabling modification of captured data packets,
- program code configured to decline a captured data packet from
processing, if said captured data packet is already processed and
modification commands induced by said captured data packet are
already determined and maintained in said list of modification
commands,
- program code configured to modify said captured data packet
based on said list of modification commands, and
- program code configured to release the modified captured data
packet.

1 40. [previously amended] A storage medium carrying a computer-
2 executable software entity for handling data packets, said data packets
3 belonging to a set of data packets having at least partly hierarchical structure,
4 said software entity comprising:

5 - program code configured to capture data packets,

6 - program code configured to accept a captured data packet for
7 processing or declining a captured data packet from processing based
8 on said captured data packet and data packets captured prior to said
9 captured data packet such that data packets are accepted for
10 processing in the order specified by said at least partly hierarchical
11 structure

12 - program code configured to decline a captured data packet from
13 processing, if a data packet immediately preceding said captured data
14 packet in said at least partly hierarchical structure is not yet captured,

15 - program code configured to delay data packets declined from
16 processing,

17 - program code configured to accept a delayed data packet for
18 processing and processing said delayed data packet, if data packet(s)
19 immediately preceding said delayed data packet in said at least partly
20 hierarchical structure is(are) already processed, and

21 - program code configured to release the delayed data packet.

1 41. [previously amended] A storage medium carrying a computer-executable
2 software entity for handling data packets, said data packets belonging to a set
3 of data packets and forming a plurality of groups of data packets, said software
4 entity comprising:

5 - program code configured to capture data packets,

6 - program code configured to decline a captured data packet
7 belonging to a first group of data packets from processing if all other
8 data packets belonging to said first group of data packets are not yet
9 captured, and delaying said captured data packet, and

10 - program code configured to process a captured data packet
11 belonging to a first group of data packets together with delayed data
12 packets belonging to said first group of data packets, if said captured
13 data packet belonging to said first group and said delayed data packets
14 belonging to said first group form a full first group of data packets.

1 42. [previously amended] A storage medium carrying a computer-executable
2 software entity for handling data packets, said data packets belonging to a set
3 of data packets and forming a plurality of groups of data packets, said software
4 entity comprising:

5 - program code configured to capture data packets, said data

6 packets containing information fragments belonging to a sequence of
7 information fragments,

8 - program code configured to decline a captured data packet from
9 processing, if a data packet containing the information fragment
10 immediately preceding the information fragments of said captured data
11 packet in said sequence is not yet captured, and

12 - program code configured to accept a captured data packet for
13 processing if a data packet containing the information fragment
14 immediately preceding the information fragments of said captured data
15 packet in said sequence is already processed, whereby each
16 information fragment of said sequence is processed only once.

1 43. [previously amended] A network element for handling data packets,
2 comprising

3 - program code configuring the network element to capture data
4 packets, said data packets belonging to a set of data packets,

5 - program code configuring the network element to accept a
6 captured data packet for processing based on said captured data packet
7 and data packets captured prior to said captured data packet,

8 - program code configuring the network element to, when
9 processing at least one captured data packet accepted for processing,

10 determine a modification command affecting at least said at least one
11 captured data packet,
12 - program code configuring the network element to maintain a list
13 of modification commands, said list enabling modification of captured
14 data packets,
15 - program code configuring the network element to decline a
16 captured data packet from processing if said captured data packet is
17 already processed and modification commands induced by said
18 captured data packet are already determined and maintained in said list
19 of modification commands,
20 - program code configuring the network element to modify said
21 captured data packet based on said list of modification commands, and
22 - program code configuring the network element to release the
23 modified captured data packet.

1 44. [previously amended] A network element for handling data packets,
2 comprising:
3 - program code configuring the network element to capture data
4 packets, said data packets belonging to a set of data packets having at
5 least partly hierarchical structure,
6 - program code configuring the network element to accept a

7 captured data packet for processing or declining a captured data packet
8 from processing based on said captured data packet and data packets
9 captured prior to said captured data packet such that data packets are
10 accepted for processing in the order specified by said at least partly
11 hierarchical structure

- 12 - program code configuring the network element to decline a
13 captured data packet from processing if a data packet immediately
14 preceding said captured data packet in said at least partly hierarchical
15 structure is not yet captured,
- 16 - program code configuring the network element to delay data
17 packets declined from processing,
- 18 - program code configuring the network element to accept a
19 delayed data packet for processing and processing said delayed data
20 packet if data packet(s) immediately preceding said delayed data packet
21 in said at least partly hierarchical structure is(are) already processed,
22 and
- 23 - program code configuring the network element to release the
24 delayed data packet.

1 45. [previously amended] A network element for handling data packets,
2 comprising

3 - program code configuring the network element to capture data
4 packets, said data packets belonging to a set of data packets and
5 forming a plurality of groups of data packets,

6 - program code configuring the network element to decline a
7 captured data packet belonging to a first group of data packets from
8 processing, if all other data packets belonging to said first group of data
9 packets are not yet captured, and delaying said captured data packet,
10 and

11 - program code configuring the network element to process a
12 captured data packet belonging to a first group of data packets together
13 with delayed data packets belonging to said first group of data packets, if
14 said captured data packet belonging to said first group and said delayed
15 data packets belonging to said first group form a full first group of data
16 packets.

1 46. [previously amended] A network element for handling data packets,
2 comprising:

3 - program code configuring the network element to capture data
4 packets, said data packets belonging to a set of data packets and
5 forming a plurality of groups of data packets and containing information
6 fragments belonging to a sequence of information fragments,

7 - program code configuring the network element to decline a
8 captured data packet from processing, if a data packet containing the
9 information fragment immediately preceding the information fragments
10 of said captured data packet in said sequence is not yet captured, and
11 - program code configuring the network element to accept a
12 captured data packet for processing if a data packet containing the
13 information fragment immediately preceding the information fragments
14 of said captured data packet in said sequence is already processed,
15 whereby each information fragment of said sequence is processed only
16 once.